

ALEA

Atelier logiciel libre pour la modélisation des plantes

Christophe Pradal, Christophe Godin



Modélisation des plantes

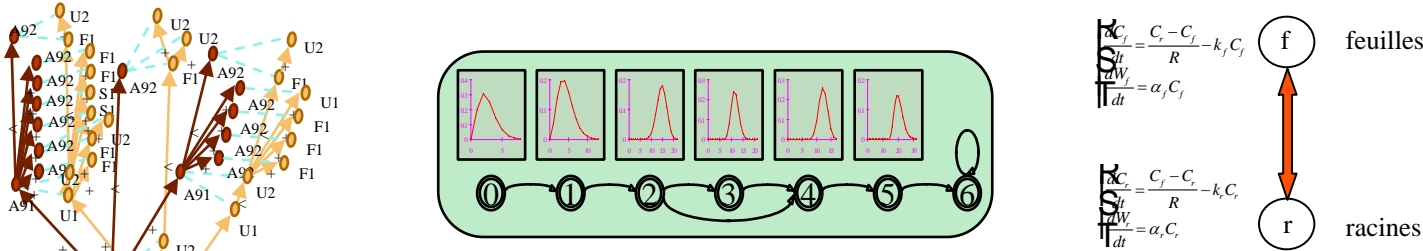
Botanique:



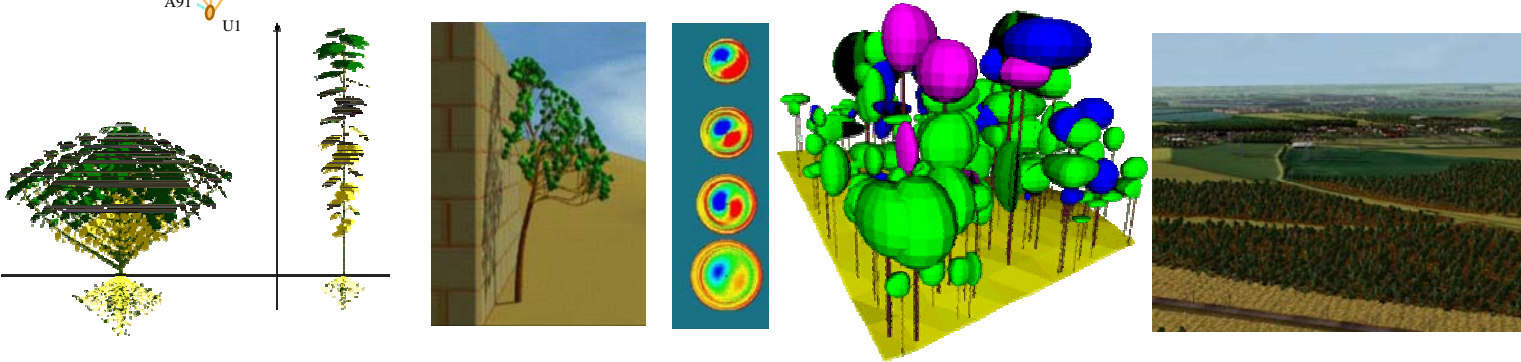
Mesure:



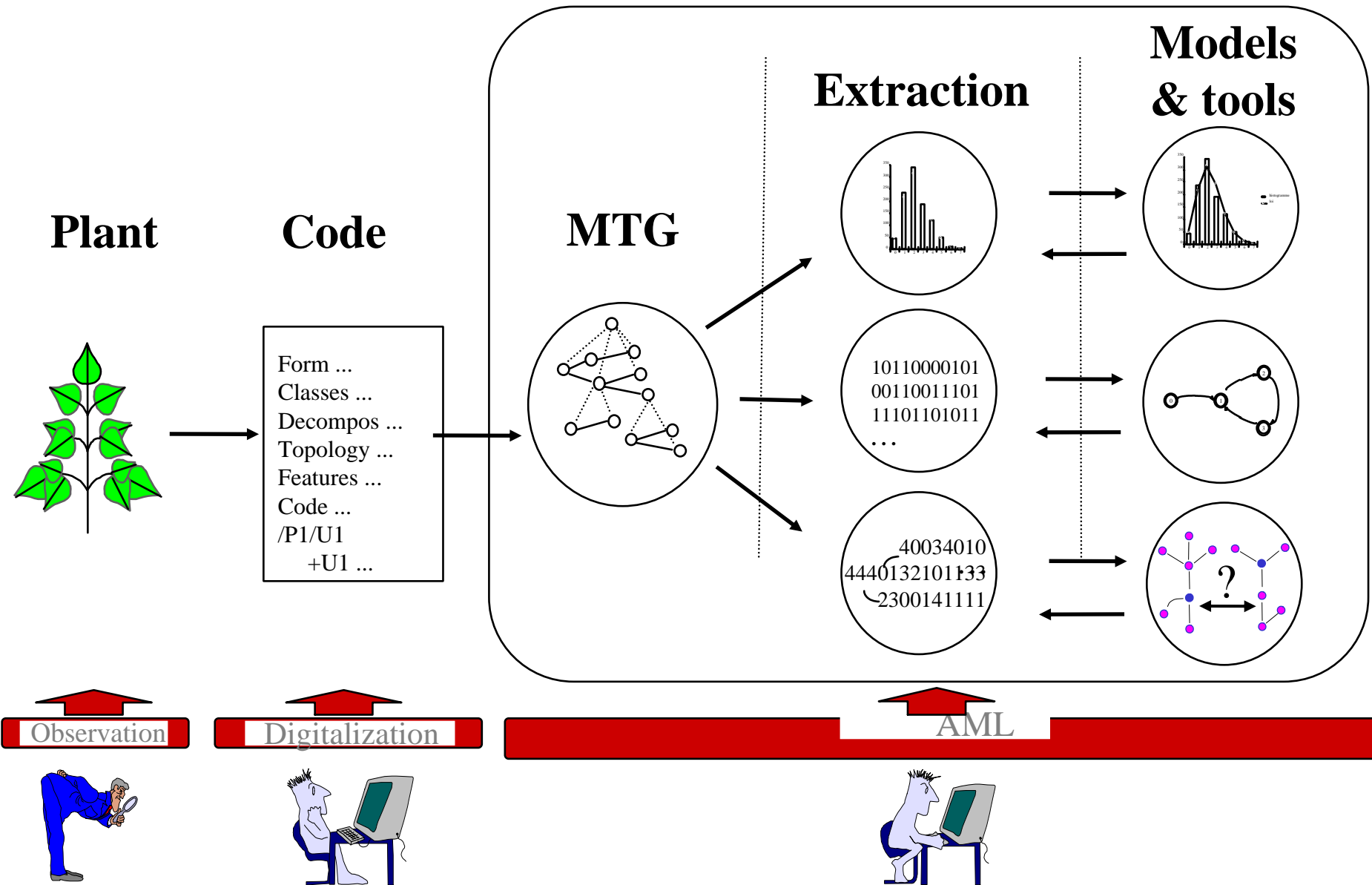
Modélisation:



Simulation:



AMAPmod - 1995



AMAPmod - 1995

Analyse et modélisation de l'architecture des plantes

- Développé par deux chercheurs en collaboration avec des biologistes.
- 400 kl de C++ étendues par le langage «maison» AML

Goulot d'étranglement

- Demandes des biologistes en application
- Formation
- Maintenance
- Nouveaux développements



Choix Logiciel libre

- Licence GPL
- Des utilisateurs dans plus de 20 pays
- 150 chercheurs formés, 8 formations
- Architecture statique, sentiment de propriété => pas de co-développement

FSPM - 1996

FSPM (Functional / Structural Plant Modelling)

- Communauté scientifique international de modélisation des plantes (workshops, ~200 chercheurs)

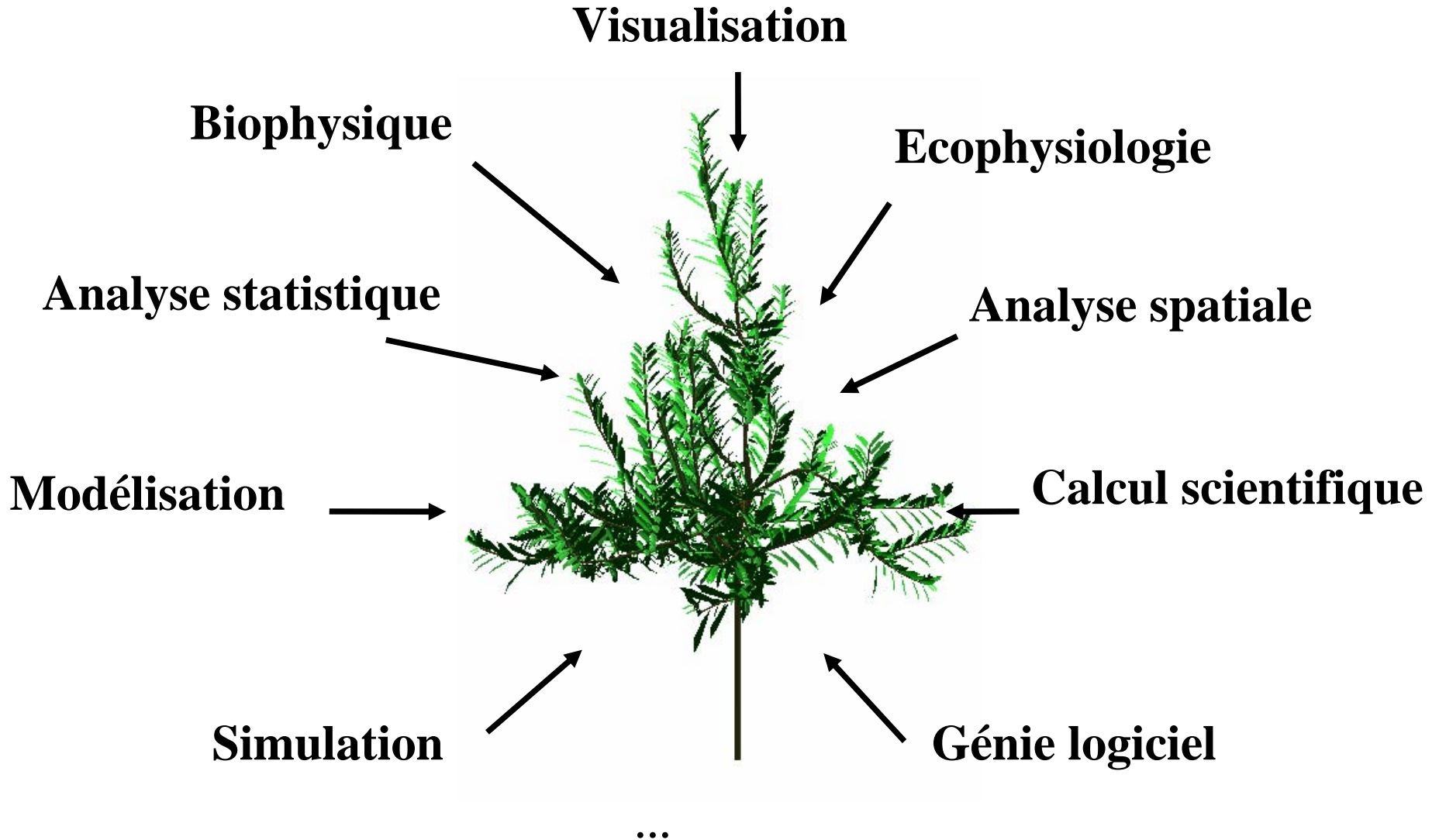
Modélisation de la croissance et du fonctionnement des plantes

- Représentation 3D de l'architecture des plantes
- Modèles physiologiques et environnementaux
- Outils d'analyse et de simulation

Domaines d'application

- Botanique, foresterie, agronomie et écologie

Un domaine pluridisciplinaire



Réseau PAIS - 2000

Plant **A**rchitecture **I**nformation **S**ystem

- Réseau européen et international

Objectifs

- Partage des bases de données
- Définir un standard de representation des plantes
- Diffuser modèles et outils logiciels
- Formation

Limites

- Pas d'homogénéité dans la démarche logicielle
- Pas de règles de diffusion

Atelier Logiciel pour la modélisation Ecophysiologique de l'Architecture des plantes

- Utiliser, évaluer et construire des modèles expérimentaux.

Objectifs

- Communauté autour d'un logiciel
 - Co-développement entre équipes de recherche
 - Echange d'expérience et d'outils
 - Formation
- Architecture à composants logiciels
 - Intégration des modèles existants (C, C++, Fortran)
 - Développement rapide de nouveaux modèles
 - Qualité logicielle pour diffusion et maintenance

Partenaires

- INRA, CIRAD, INRIA, LABRI.

Logiciel Libre - Intérêts scientifiques

Validation scientifique

- Accès code source => Vérification des modèles, algos, outils.

Collaboration scientifique

- Les modèles publiés sont directement accessibles.

Accessible et adaptable par tous

- Code adapté par chacun en fonction de ses besoins.
- Modèles compatibles et comparables.

Pérennité du code

- En cas d'arrêt du développement, quiconque peut reprendre un projet et continuer à le développer.

Protection du droit d'auteur

- Licence libre

Logiciel Libre – Avantage pratique

Synergie

- Collaboration entre développeurs (souvent isolés)
- Collaboration entre équipes pluridisciplinaires

Economie d'échelle

- Développement
- Diffusion et Maintenance
- Formation
- Veille technologique

Qualité

- Rationalisation des choix
- Règles communes

Communauté libre ALEA

Architectes

- Développent le noyau principale du système
- Conseil, formation et assistance auprès des modélisateurs

Modélisateurs

- Développent et intègrent leurs modules
- Conseil, formation et assistance auprès des utilisateurs

Utilisateurs

- Définissent des scénarios
- Mettent à disposition des bases de données

Communauté libre

- Licence libre du noyau: LGPL
- Critères de diffusions:
 - Copyright & license.
 - Qualité logicielle (GUI, tests, documentation)

Logiciel Libre – ALEA

Communauté libre

- Développeurs, modélisateurs et utilisateurs
- Règles de fonctionnement (Charte ALEA)

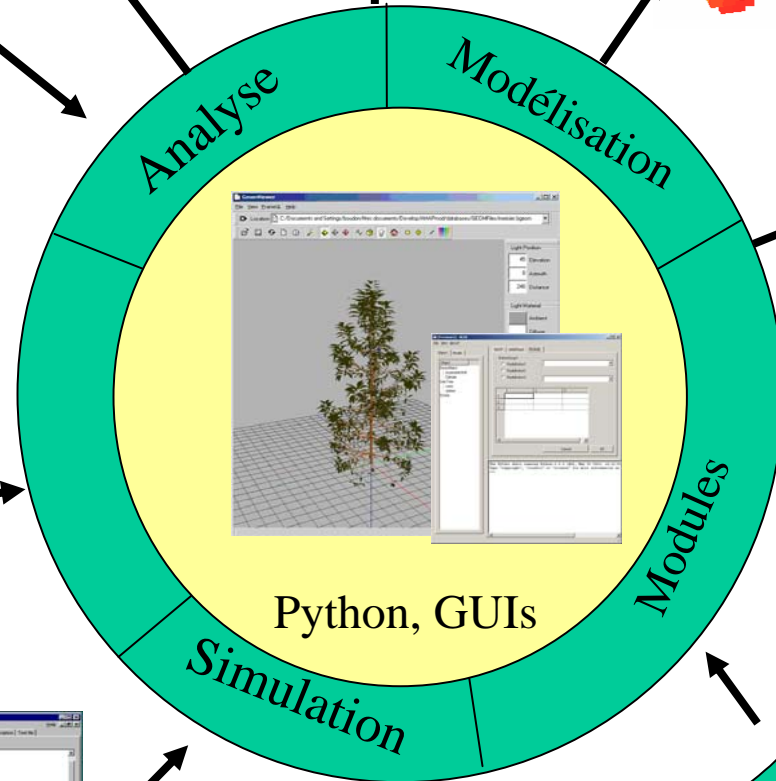
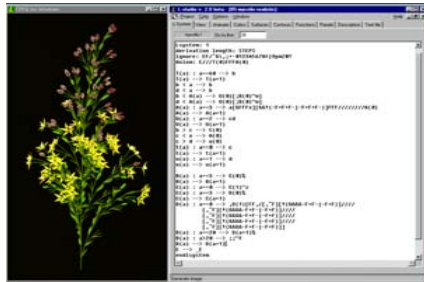
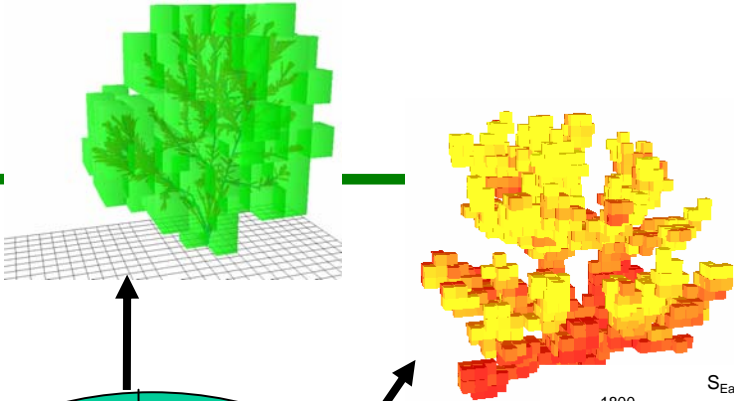
Licence libre (LGPL)

Animation de la communauté

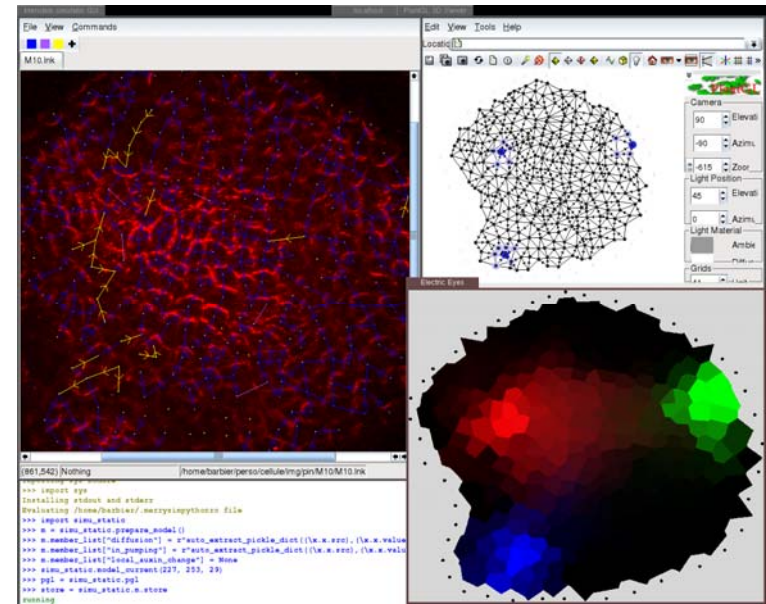
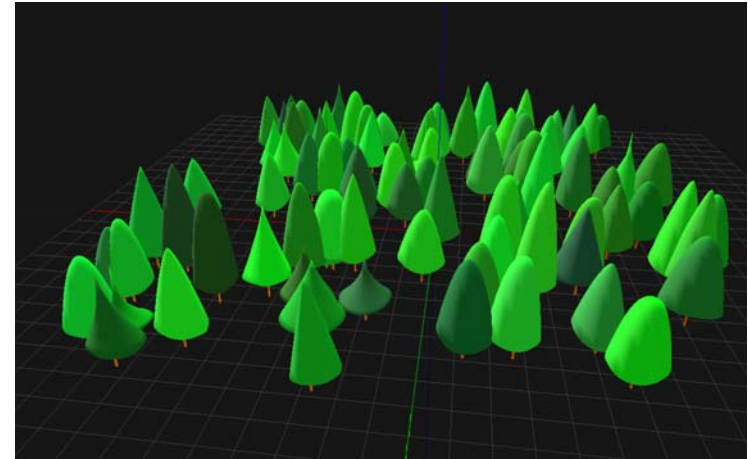
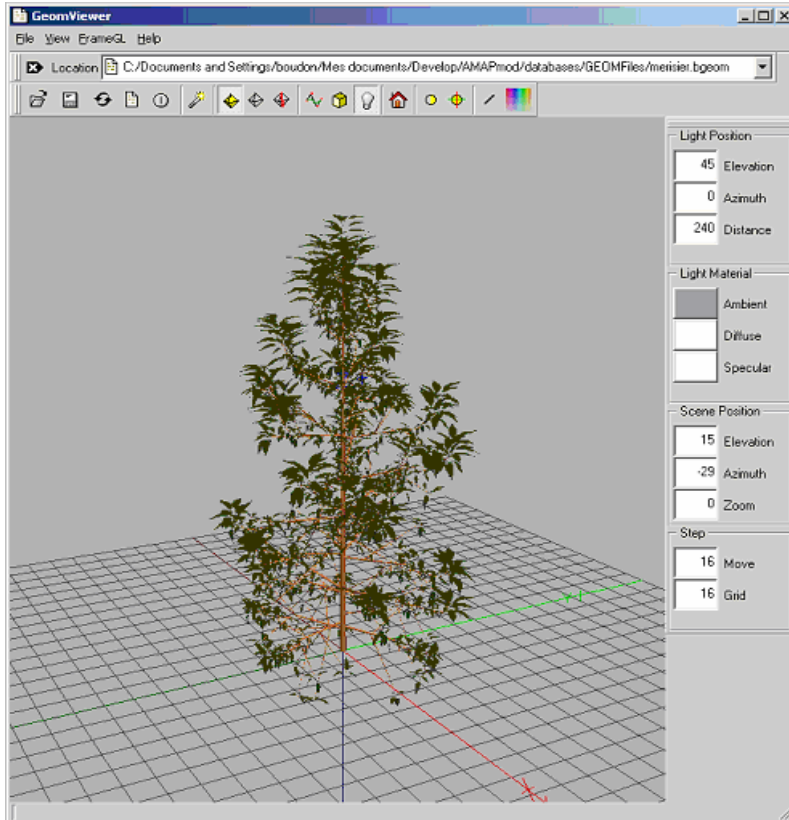
- Réunions régulières des architectes
- 2 formations des modélisateurs
- 3 formations des utilisateurs (France, Brésil, Thaïlande)

Développement collaboratif

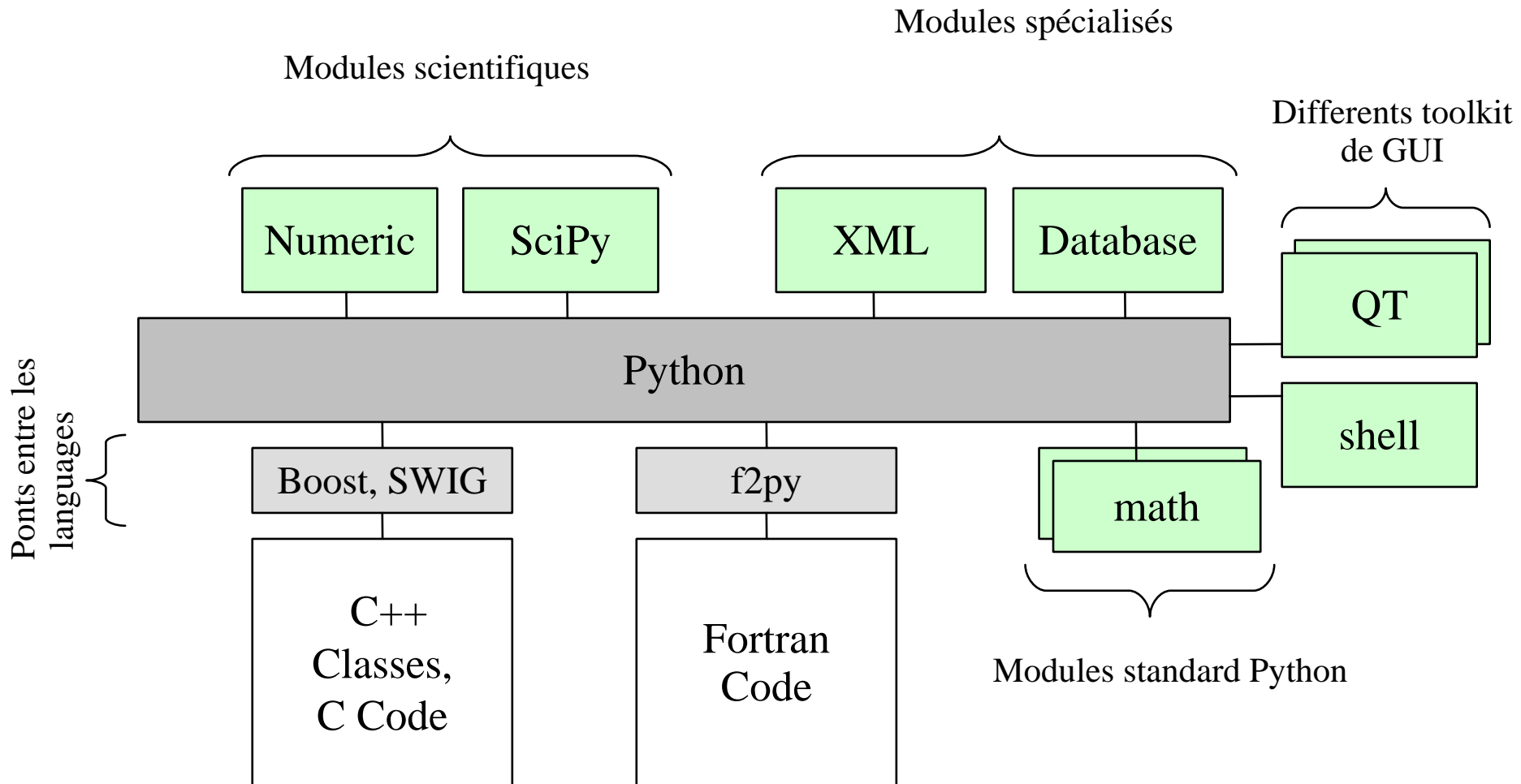
- CVS, Forums
- Site web (en construction)



ALEA: différents objets biologiques



Python vu comme un Bus Logiciel



ALEA – Description technique

- ✓ **Langages:** C, C++, Fortran et Python
- ✓ **OS:** Gnu/Linux, Windows
- ✓ **Logiciels utilisés**
 - ✓ **GUI:** QT et PyQt
 - ✓ **Gestion de versions:** CVS
 - ✓ **Documentation:** Doxygen (C, C++), epydoc (Python)
 - ✓ **Tests unitaires:** cppUnit, PyUnit
 - ✓ **Compilation:** SCons
 - ✓ **Wrappers:** Boost.Python, Swig et f2py

Logiciel Libre – Tension liée au don

Le don perçu comme une perte

- L'utilisateur n'est plus captif.
- Comment valoriser et financer le coût de développement?

Le don comme un échange

- Partage des coûts de développement des algos, de la formation et de la diffusion du logiciel.
- Partage des connaissances
 - Scientifiques
 - Techniques (génie logiciel)
 - Bases de Données (plantas)

ALEA – Bilan

Logiciel Libre, un modèle adapté pour la recherche

- Création d'une communauté libre française entre équipes initialement concurrentes.
 - => Développement d'un outil de collaboration
 - => Propriété partagée
 - => Partage des connaissances entre équipes
 - => Amélioration de la qualité
- Co-développement possible

Un bénéfice pas encore unanimement reconnu

- Perte du contrôle total
- Problème d'identité de l'auteur
- Coût de la qualité
- La transparence fait souvent peur
- Valorisation par la publication

Questions ouvertes

Comment financer le développement logiciel ?

- Double licence ?
- Consortium ?
- ... ?

Comment valoriser l'activité de production logicielle ?

- Investissement lourd
- Faible reconnaissance des instances d'évaluation
- Logiciel = production scientifique ?

Un modèle pour le partage des données en biologie ?
